

REMARKS

Claims 1 to 15 were pending in the present application. Applicant has amended claims 1, 2, and 6, canceled claims 3 to 5, 7, and 9 to 15, and added claim 16. Claims 1, 2, 6, 8, and 16 remain pending.

Objections to the Drawings

The Examiner objected to the drawings under 37 C.F.R. § 1.83(a) because it does not show each child node having pointers to previous and next children nodes as claimed. Applicant has amended the claims to recite that each child node has at least one of (1) a pointer that points to a previous child node and (2) a pointer that points to a next child node. Accordingly, Applicant respectfully requests the Examiner to withdraw the objections to the drawings.

Claim Objections

The Examiner objected to claims 3 and 5 for informalities. Applicant has canceled claims 3 and 5, thereby rendering their rejections moot.

§ 101 Rejections

The Examiner rejected claims 1 to 15 under 35 U.S.C. § 101 because the claimed invention lacks patentable utility. Applicant has amended independent claims 1 and 6 to recite a “data structure for data storage in a machine readable medium.” Applicant notes that data structures are commonly used for data storage in databases and file systems.

Claim 2 depends from amended claim 1, and claims 8 and 16 depend from amended claim 8. Applicant has canceled claims 3 to 5, 7, and 9 to 15, thereby rendering their rejections moot. Accordingly, Applicant respectfully requests the Examiner to withdraw the § 101 rejections.

§ 112 Rejections

The Examiner rejected claims 1 and 2 under 35 U.S.C. § 112, first paragraph, for failing to comply with the enablement requirement.

The Examiner found that the specification does not support claim 1, which recites that each child node has pointers to previous and next children nodes. As discussed above, Applicant has

amended claim 1 to recite that each child node has at least one of (1) a pointer that points to a previous child node and (2) a pointer that points to a next child node.

The Examiner also found that the specification does not include a clear and concise description of the manner and the process for querying the data structure and subsequently providing a pointer from a parent node to a child node that was queried. Applicant respectfully traverses and submits that one skilled in the art would know how to query the data structure and how the invention works in view of the specification as described below.

Paragraphs [0010] to [0012] of the specification provide that common children nodes from a parent node are linked in a predetermined order. As can be seen from Fig. 2, children nodes 54-0 to 54-N descend from parent node 52, grandchildren nodes 58-0 to 58-N descend from child node 54-2, and the basic structure can repeat for additional levels. As such, each node can be referenced by nodal positions and levels from the node itself up to root node 52. Thus, node 58-2 can be referenced as the 3rd node in the 2nd level, which descends from the 3rd node in the 1st level (i.e., node 54-2), which descends from the root node (i.e., node 52). This is similar to the indexing described in U.S. Patent No. 6,185,569 ("East et al."). This type of referencing because each subdivision of children nodes from a parent node are arranged in the predetermined order.

To query node 58-2, we use the above information to traverse data structure 50. For example, we know that we have to go from root node 52 to node 54-2, and from node 54-2 to node 58-4. To save space, we do not have pointers that point from root node 52 to all of its children nodes. Instead, we only have pointers pHead, pTail, and pCursor. Assuming for now that pCursor is null, we look to pointers pHead and pTail to reach the nodes in the 1st level. As we know node 54-2 is the 3rd node in the 1st level, we select the pointer that brings us closer to node 54-2 for faster access. For example, if N is 19 (so there are twenty children nodes descending from root node 52), we would select pointer pHead since it brings us the closer to node 54-2 than pointer pTail. Once at node 54-0, we know we have to go to the 3rd node at the 1st level. Thus, we follow pointers pNext over the next two nodes to reach node 54-2.

Again assuming for now that pCursor is null, we look to pointers pHead and pTail to reach the nodes in the 2nd level. As we know node 58-2 is the 3rd node in the 2nd level, we again select pointer pHead since it brings us closer to node 58-2 than pointer pTail. Once at node 58-0, we know we have to go to the 3rd node at the 2nd level. Thus, we follow pointers pNext over the next two nodes to reach node 58-2.

To speed up the search the next time around, pointers pCursor at the 1st and the 2nd levels are updated so they point to the children nodes last traversed in their respective subdivisions of children nodes in their respective levels. Thus, we update pointers pCursor at the 1st and the 2nd levels to point to nodes 54-2 and 58-2, respectively. As searches are often sequential, the query of node 58-3 would be sped up tremendously since we would simply follow pointers pCursor from root node 52 to node 54-2 at the 1st level, from node 54-2 to node 58-2 at the 2nd level, and then traverse one node over to reach node 58-3.

Although a specific way of traversing data structure 50 is described, the claimed invention should not be limited by this description as there are other ways to index and traverse data trees.

The Examiner further found that the metes and bounds of the invention cannot be determined because the definition of “order” is unclear and when the claimed invention concerns physical memory locations. As described above, the order refers to a predetermined order in which common children node from a parent node are linked according to pointers pNext and pPreview. Applicant has amended the claims to clarify the “order” as the sequence in which the children nodes are linked to each other.

Accordingly, Applicant respectfully requests the Examiner to withdraw the § 112 rejections.

§ 102 Rejections

The Examiner rejected claims 1 to 15 under 35 U.S.C. § 102(b) as being anticipated by East et al. Applicant has amended claim 1, which now recites:

1. A data structure for data storage in a machine readable medium, the data structure comprising a parent node and a plurality of children nodes of the parent node, wherein:

the parent node comprises:

a first pointer always pointing to a child node that was last traversed in data access;

the children node are linked in an order where each child node comprises at least one of:

a second pointer pointing to a next child node in the order; and

a third pointer pointing to a previous child node in the order.

Amended claim 1 (emphasis added). As described above with regards to the § 112 rejections, a pointer of a parent node is updated after data access so that pointer always points to the last traversed child node in the data access. East et al. does not disclose a pointer that always points to a child node that was last traversed in data access. Instead, East et al. appears to disclose an inner child pointer that fixedly points to a child node and is not updated after data access. Accordingly, amended claim 1 is patentable over East et al.

Claim 2 depends from amended claim 1 and is patentable over East et al. for at least the same reasons as amended claim 1.

Applicant has canceled claims 3 to 5, thereby rendering their rejections moot.

Applicant has amended claim 6, which now recites:

6. A method for creating a data structure for data storage in a machine readable medium, the data structure comprising a parent node and a plurality of children nodes of the parent node, the method comprising:

for the parent node, creating:

first and second pointers pointing to two of the children nodes; and

a third pointer for pointing to a last traversed child node in data access;

linking the children node in an order, comprising, creating for each child node at least one of:

a fourth pointer pointing to a next child node in the order; and

a fifth pointer pointing to a previous child node in the order;

accessing a data from the data structure by traversing at least one of the children nodes; and

for the parent node, updating the third pointer to point to the last traversed child node in said accessing a data.

Amended claim 6 (emphasis added). As discussed above, East et al. does not disclose updating a pointer of a parent node after data access so that pointer always points to the last traversed child node in the data access. Accordingly, amended claim 6 is patentable over East et al.

Claims 8 and 16 depend from amended claim 6 and are patentable over East et al. for at least the same reasons as amended claim 6.

Applicant has canceled claims 7 and 9 to 16, thereby rendering their rejections moot.

Summary

In summary, claims 1 to 15 were pending in the above-identified application when last examined. Applicant has amended claims 1, 2, 6, canceled claims 3 to 5, 7, and 9 to 15, and added claim 16. For the above reasons, Applicant respectfully requests the allowance of claims 1, 2, 6, 8, and 16. Should the Examiner have any questions, please call the undersigned at (408) 382-0480.

Respectfully submitted,

/David C Hsia/

David C. Hsia
Attorney for Applicant(s)
Reg. No. 46,235

Patent Law Group LLP
2635 North First St., Ste. 223
San Jose, California 95134
408-382-0480x206